MAR 2 6 2007

IN THE CLAIMS

The current claims follow. For claims not marked as amended in this response, any difference in the claims below and the previous state of the claims is unintentional and in the nature of a typographical error.

1-25. Cancelled.

26. (Currently Amended) A method for detecting corruption associated with a stack in a storage device, the stack encompassing a range of memory of a fixed size, the method comprising the steps of:

storing a first predetermined value in a first address location immediately preceding the range of memory;

detecting the occurrence of a stack operation within the stack;~~and~~

comparing a value in the first address location to the first predetermined value to determine if the stack operation corrupted the first predetermined value stored in the first address location; and

where the stack operation is determined to have corrupted the first predetermined value, restoring the first predetermined value to the first address location.

27. (Previously Presented) The method as set forth in Claim 26, wherein the first predetermined value comprises a known bit pattern.

28. (Previously Presented) The method as set forth in Claim 26, wherein the first predetermined value comprises a processor readable address.

29. (Previously Presented) The method as set forth in Claim 26, wherein the first predetermined value comprises a processor readable instruction.

30. (Previously Presented) The method as set forth in Claim 26, wherein the stack operation inserts data in the stack.

31. (Previously Presented) The method as set forth in Claim 26, wherein the stack operation removes data from the stack.

32. (Currently Amended) The method as set forth in Claim 26, further comprising the ~~step~~ steps of:

storing a second predetermined value in a second address location immediately following the range of memory;

comparing a value in the second address location to the second predetermined value to determine if the stack operation corrupted the second predetermined value stored in the second address location; and

where the stack operation is determined to have corrupted the second predetermined value, restoring the second predetermined value to the second address location.

33. (Previously Presented) The method as set forth in Claim 32, wherein the second predetermined value comprises a known bit pattern.

34. (Previously Presented) The method as set forth in Claim 32, wherein the second predetermined value comprises a processor readable address.

35. (Previously Presented) The method as set forth in Claim 32, wherein the second predetermined value comprises a processor readable instruction.

36. (Previously Presented) The method as set forth in Claim 32, wherein the stack operation inserts data in the stack.

37. (Previously Presented) The method as set forth in Claim 32, wherein the stack operation removes data from the stack.

38. (Currently Amended) A system for detecting corruption associated with a stack, the system comprising:

a processor; and

a storage medium comprising a stack, the stack encompassing a range of memory of a fixed size, wherein the processor is operable to:

store a first predetermined value in a first address location immediately preceding the range of memory[[,]];

detect the occurrence of a stack operation within the stack; ~~and~~

compare a value in the first address location to the first predetermined value to determine if the stack operation corrupted the first predetermined value stored in the first address location; and

where the stack operation is determined to have corrupted the first predetermined value, restoring the first predetermined value to the first address location.

39. (Previously Presented) The system as set forth in Claim 38, wherein the first predetermined value comprises a known bit pattern.

40. (Previously Presented) The system as set forth in Claim 38, wherein the first predetermined value comprises a processor readable address.

41. (Previously Presented) The system as set forth in Claim 38, wherein the first predetermined value comprises a processor readable instruction.

42. (Previously Presented) The system as set forth in Claim 38, wherein the stack operation inserts data in the stack.

43. (Previously Presented) The system as set forth in Claim 38, wherein the stack operation removes data from the stack.

44. (Currently Amended) The system as set forth in Claim 38, wherein the processor is further operable to:

store a second predetermined value in a second address location immediately following the range of memory;

compare a value in the second address location to the second predetermined value to determine if the stack operation corrupted the second predetermined value stored in the second address location; and

where the stack operation is determined to have corrupted the second predetermined value, restoring the second predetermined value to the second address location.

45. (Previously Presented) The system as set forth in Claim 44, wherein the second predetermined value comprises a known bit pattern.

46. (Previously Presented) The system as set forth in Claim 44, wherein the second predetermined value comprises a processor readable address.

47. (Previously Presented) The system as set forth in Claim 44, wherein the second predetermined value comprises a processor readable instruction.

48. (Previously Presented) The system as set forth in Claim 44, wherein the stack operation inserts data in the stack.

49. (Previously Presented) The system as set forth in Claim 44, wherein the stack operation removes data from the stack.